

DockerコンテナでのOracle WebLogic Serverの運用

Oracleホワイト・ペーパー | 2020年4月



ORACLE®

目次

DockerでのOracle WebLogic Serverの運用	2
Oracle WebLogic Server Dockerイメージ	2
カスタム・ビルドのOracle WebLogic Serverイメージ	3
GitHubのDockerfileとスクリプト	3
Dockerfile	3
スクリプト	5
DockerコンテナでのOracle WebLogic Serverのクラスタリング	5
ビルドと実行の方法	7
Dockerfile	7
スクリプト	7
Oracle WebLogic Serverイメージのビルド	8
Oracle WebLogic Server ドメイン作成用サンプル	9
Oracle WebLogic Serverのサンプル・ドメイン	9
Oracle WebLogic Server ドメインのサンプルDockerイメージのビルド	10
Oracle WebLogic Server管理サーバー・コンテナの実行	11
Oracle WebLogic Server管理対象サーバー・コンテナの実行	12
その他のサンプル	13
Oracle WebLogic ServerをDockerで実行する場合の考慮事項	13
まとめ	15

DockerでのOracle WebLogic Serverの運用

Oracle WebLogic Server (Oracle WLS) はDockerコンテナおよびCRI-Oコンテナ上で正しく動作することが保証されています。[GitHub](#)でOracle WebLogic ServerのイメージをビルドするためのDockerfileとサポート・スクリプトが公開されています。これらのイメージは、既存のOracle Linuxイメージの拡張版です。ユーザーは、これらのOracle WLS Dockerイメージを使用することも、独自のイメージを作成することもできます。

Dockerは、分散アプリケーションのビルド、パッケージ化、配信、実行ができるプラットフォームです。ユーザーはDockerを使用して、アプリケーションとその依存先のライブラリまたはファイルをパッケージ化し、1つのDockerイメージにまとめます。

Dockerイメージは移植可能なアーティファクトであり、複数のLinux環境に分散させることができます。配布されたイメージを使用してコンテナをインスタンス化すると、同じホスト・オペレーティング・システム上の別のコンテナで実行されている別のアプリケーションとは独立した状態で、そのコンテナ上でアプリケーションを実行できます。

Oracle WebLogic Serverのバージョン12.1.3、12.2.1.3.0、12.2.1.4.0、14.1.1.0.0は、DockerコンテナおよびCRI-O上で正しく動作することが保証されています。Dockerイメージを作成する際には、Oracle WebLogic Server、JDK、Linux、Dockerのバージョンを組合せて利用できます。サポートされる最新のOracle WebLogic Server構成の詳細およびサポートに関する声明については、[Oracle Fusion Middlewareの動作保証状況に関するページ](#)を参照してください。

提供されたDockerfileとスクリプトを使用すれば、Oracle WLSドメインのクラスタ構成または非クラスタ構成を作成できます。作成されたドメイン構成で稼働する各サーバーは、それぞれのDockerコンテナ内で稼働し、他のサーバーと通信できます。他の構成やアプローチも可能です。他の構成の作成方法をこのホワイト・ペーパーで説明します。

Oracle WebLogic Server Dockerイメージ

[GitHub](#)にあるDockerfileとサポート・スクリプトを使用して、Oracle WebLogic Server Dockerイメージがビルドされます。これらのイメージは、既存のOracle Linux 7.0スリム・イメージの拡張版であり、Server JRE 7、8、またはJDK 11と、Oracle WebLogic Server 12c (12.1.3、12.2.1.3、12.2.1.4) および14c (14.1.1.0) のインストールが含まれます。

公開されているイメージは以下の3種類です。

1. Oracle WebLogic Server汎用イメージ
2. Oracle WebLogic Server開発者向けイメージ
3. Oracle WebLogic Serverスリム・イメージ

Oracle WebLogic Server汎用イメージは“汎用インストーラ”を使用してビルドされたもので、Oracle WebLogic ServerとOracle Coherenceの全ソフトウェア（サンプル付き）が含まれ、あらゆる開発目的および本番目的でサポートされます。Oracle WebLogic Server開発者向けイメージは“クイック・インストーラ”を使用してビルドされたもので、開発目的を想定しています。

これには、Oracle WebLogic ServerとOracle Coherenceのランタイム・ソフトウェアがすべて含まれますが、サンプルおよびOracle WebLogic Server管理コンソールのローカライズ版ヘルプ・ファイルは含まれていません。Oracle WebLogic Serverスリム・イメージは“スリム・インストーラ”を使用してビルドされたものであり、Oracle WLSコンソールによる監視と構成が不要な場合の、DockerまたはCRI-OイメージおよびKubernetes内のコンテナを使用する開発および本番での用途を想定しています。これには、Oracle WebLogic ServerとOracle Coherenceサーバーのランタイム・ソフトウェアがすべて含まれますが、サンプル、Oracle WebLogic Server管理コンソール、Oracle WebLogicクライアント、Mavenプラグイン、およびJava DBは含まれていません。

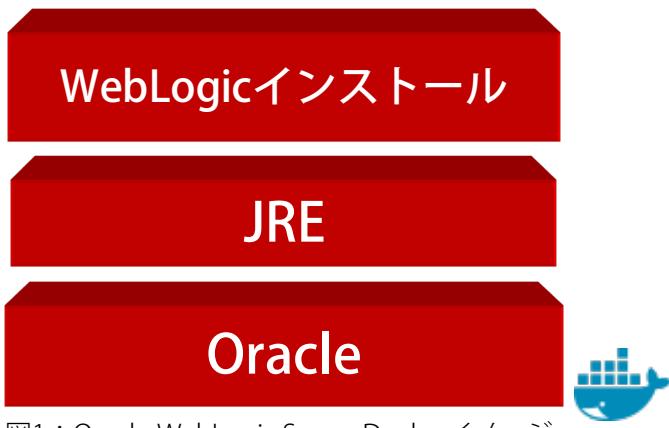


図1：Oracle WebLogic Server Dockerイメージ

カスタム・ビルドのOracle WebLogic Serverイメージ

Oracle WebLogic Server Dockerイメージは独自に作成することができます。初期段階のサンプルとして使用できるDockerfileとスクリプトを[GitHubのOracle WebLogic Server Dockerfile](#)で公開しています。

カスタムのOracle WLSイメージをビルドする際に必要なものは以下のとおりです。

1. Oracle Linuxのベース・イメージ
2. GitHubから入手したDockerfileとスクリプト
3. Oracle WebLogic Server汎用インストーラ、クイック・インストーラ、またはスリム・インストーラ
4. 対応するServer JREまたはJDK

Oracle WebLogic ServerイメージはOracle WebLogic Image Toolを使用して作成し、パッチを適用することができます。詳しくは、[Image ToolのGitHub](#)プロジェクトを参照してください。

GitHubのDockerfileとスクリプト

Dockerfile

Oracle WebLogic Serverのインストール・イメージを作成するDockerfileでは、以下の処理が実行されます。

1. Oracle Linuxスリムのベース・イメージを使用して、Oracle WebLogic Server JREイメージを拡張してビルドします。
2. 汎用インストーラ、クイック・インストーラ、またはスリム・インストーラを使用して、Oracle WLSをサイレント・モードでインストールします。

独自のOracle WebLogic Serverインストール・イメージを作成した後、このイメージを拡張して、Oracle WebLogic Serverのベース・ドメインが構成された状態にすることができます。詳しくは、[Oracle DockerのGitHub](#)プロジェクトのサンプルを参照してください。

Oracle WebLogic Server ドメイン・イメージを作成するDockerfileでは、以下の処理が実行されます。

1. Oracle WebLogic Serverインストール・イメージを拡張します。
2. Oracle WebLogic Serverスクリプト・ツール (Oracle WLST) のスクリプトまたはOracle WebLogic Deploy Toolingを呼び出して、Oracle WLS ドメインを構成します。

このドメインには、1台の管理サーバー、複数の管理対象サーバーから成る1つのOracle WebLogicクラスタ、およびこのクラスタにデプロイされるリソース (データソース、JMSなど) が含まれます。

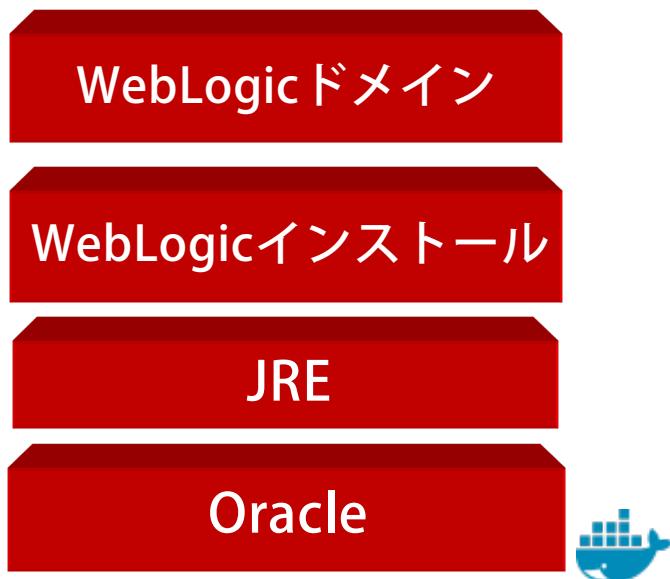


図2：Oracle WebLogic Server ドメイン・イメージ

このOracle WebLogic Server ドメイン・イメージを使用して、以下の2種類のコンテナを作成できます。

- Oracle WebLogic Server管理サーバー1台を含む管理サーバー・コンテナ
- 管理対象サーバー・コンテナ



図3：Oracle WebLogic Serverコンテナの種類

スクリプト

スクリプトはOracle WebLogic Serverイメージを作成するために使用でき、Oracle WebLogic Server ドメインの構成によるイメージ拡張のサンプルとしても機能します。

DockerコンテナでのOracle WebLogic Serverのクラスタリング

Dockerコンテナを使用すれば、 Oracle WLS ドメインのクラスタ構成または非クラスタ構成を作成できます。 ドメイン内の各サーバーは、それぞれのDockerコンテナ内で稼働し、同じホスト上の他のサーバーと通信できます。



図4：単一ホスト内でのDockerコンテナでのOracle WebLogic Serverのクラスタリング

このトポロジのメリットは次のとおりです。

- 従来型のデプロイメントに適している
- Oracle WLS ドメイン・イメージからコンテナをデプロイするのが容易
- クラスタのスケール・アップやスケール・ダウンが容易
- 開発者向けとして適している
- ホストでのインストールや構成が必要なものはDockerのバイナリのみ

コンテナは、物理サーバーまたは仮想サーバーのLinuxホスト1台にすべてを配置することも、図5のように複数の物理サーバーまたは仮想サーバーのLinuxホストに配置することもできます。

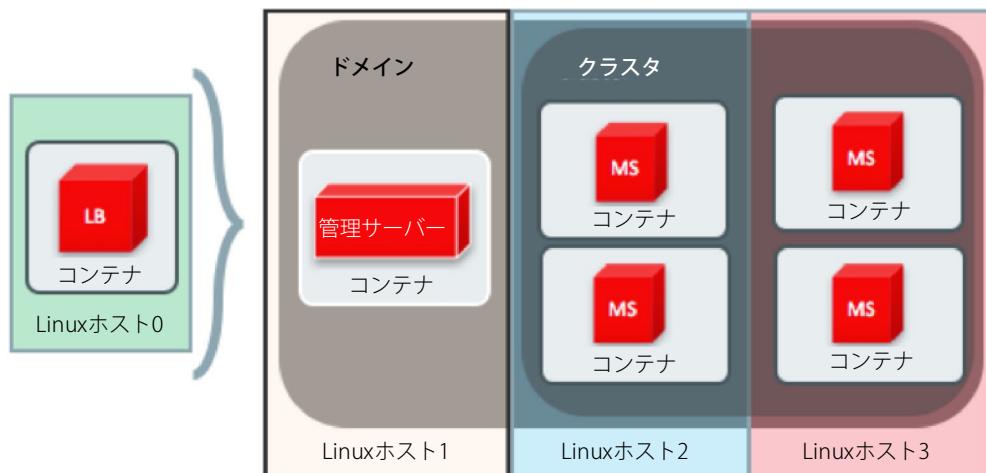


図5：複数のホスト上でコンテナ化されたOracle WebLogic Serverアプリケーション

このトポロジのメリットは次のとおりです。

- コンテナ化したアプリケーションとサービス向けの“Docker方式”
- コンテナの再生成が容易
- 各コンテナは、同じOracle WLS ドメインのインスタンス

コンテナ化したアプリケーションとサービス向けの“Docker方式”に沿ったトポロジは、すべてのリソース、共有ライブラリ、デプロイメントを含み、管理サーバーのみを実行するコンテナで構成されます。

このDockerイメージは、事前定義されたすべてのドメイン・リソース、アプリケーション、および共有ライブラリをデプロイ済みの状態で含み、管理対象サーバーまたはクラスタは構成されません（図6参照）。

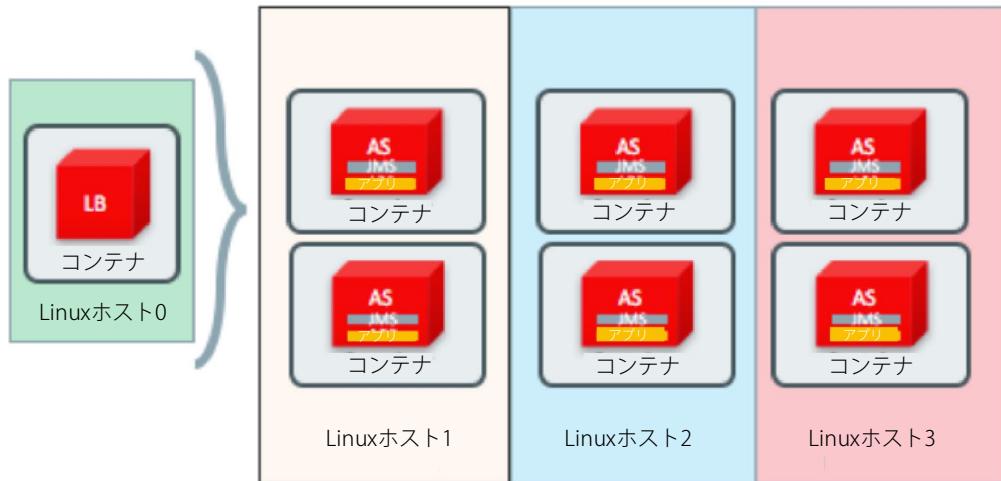


図6：複数のホスト上に構成された、アプリケーションおよびリソースを含む単一のサーバー・ドメイン

ビルドと実行の方法

Oracle WebLogic Serverインストール・イメージのビルドとこのイメージを拡張してOracle WebLogic Serverドメイン・イメージを作成するためのDockerfileとサポート・スクリプトが[GitHub](#)にあります。Oracle WebLogic Serverイメージをビルドしてコンテナを起動するには、まずこのディレクトリ構造全体をダウンロードしてください。

Dockerfile

Oracle WebLogic Server 12.2.1.4.0のDockerfileが/OracleWebLogic/dockerfiles/12.2.1.4サブディレクトリ内に3つあります。それぞれ、Oracle WLS 'developer' (開発者向け) イメージ、Oracle WLS 'generic' (汎用) イメージ、Oracle WLS 'slim' (スリム) イメージをビルドするためのDockerfileです。

Dockerfile.developer

Dockerfile.generic

Dockerfile.slim

スクリプト

/OracleWebLogic/dockerfiles/12.2.1.4/container-scriptsサブディレクトリ内にサポート・スクリプトがあります。これらは上記のDockerfileとともに、Oracle WebLogic Serverイメージをビルドし、単一のサーバー・ドメインを構成するために必要です。

スクリプト名	機能
buildDockerImage.sh	Oracle WLSインストール用Dockerfile命令を使用してOracle WLSイメージをビルドします。
container-scripts/createAndStartEmptyDomain.sh	propertiesファイルから資格証明を取得し、boot.propertiesファイルを作成し、create-wls-domain.pyを呼び出してOracle WLSTを実行してドメイン構成を作成し、管理サーバーを起動します。
container-scripts/create-wls-domain.py	1台の管理サーバーを含むベース・ドメインを構成します。
properties/domain.properties	管理サーバーの資格証明が含まれます。

Oracle WebLogic Serverイメージのビルド

このOracle WebLogicイメージはOracle Server JRE 8イメージを拡張したものです。このホワイト・ペーパーで示されるDockerfileとスクリプトを使用するには、以下のDockerリポジトリのクローンをローカル・マシンに作成します。

```
$ git clone https://github.com/oracle/docker-images.git
```

../../../../OracleJava/java8内のDockerfileを使用してイメージをビルドするか、あらかじめビルドされたイメージを[Oracle Container Registry](#)から取得します。初めて使用するユーザーは、こちらの[説明](#)に従ってください。Oracle Container Registryからイメージを取得するコマンドは以下のとおりです。

```
$ docker login container-registry.oracle.com
$ docker pull container-registry.oracle.com/java/serverjre:8
$ docker tag container-registry.oracle.com/java/serverjre:8 oracle/serverjre:8
```

まず、使用するインストールの種類（汎用インストーラ、クイック・インストーラ、スリム・インストーラ）を決定し、必要なOracle WebLogic Serverインストーラをdockerfiles/12.2.1.4フォルダにダウンロードします。dockerfilesフォルダから、buildDockerImage.shスクリプトをrootとして実行します。

```
$ sudo sh buildDockerImage.sh -h
```

使用方法：buildDockerImage.sh [-d|-g|-m]

パラメータ：

-d : 'developer'ディストリビューションに基づいてイメージを作成

-g : 'generic'ディストリビューションに基づいてイメージを作成

-m : 'slim'ディストリビューションに基づいてイメージを作成

注: 作成されるイメージでは、ドメインがあらかじめ構成されていません。別のDockerfileとサポート・スクリプトがあるので、それを使用してOracle WebLogic Serverインストール・イメージを拡張してOracle WLSドメイン・イメージを作成します。

Oracle WebLogic Server ドメイン作成用サンプル

Oracle WebLogic Serverインストール・イメージを拡張するカスタムDockerfileからドメインを作成する方法が分かるように、Oracle WebLogic Server 12cのサンプルをいくつか用意しています。Oracle WebLogic Serverのドメイン・ホームをイメージ内に保存するかホスト・ボリューム上に保存するかという観点から、異なるトポロジがサポートされています。ドメイン・ホームは、Oracle WLSTまたはOracle WebLogic Deploy Tooling (Oracle WDT) を使用して作成できます。Oracle WebLogic Deploy Toolingについて詳しくは、[Oracle WebLogic Deploy ToolingのGitHub](#)プロジェクトを参照してください。

Oracle WebLogic Serverのサンプル・ドメイン

OracleWebLogic/samples/12213-domain内のDockerfileは、oracle/weblogic:12.2.1.3-dev (Developerディストリビューション) を拡張してイメージを作成します。このDockerfileは、Oracle WebLogic Server ドメインをホスト・ボリューム上に構成します。Oracle WebLogic Server ドメインには、1台の管理サーバーと、1台以上の管理対象サーバーから成るOracle WebLogicクラスタ (構成済みまたは動的) が含まれます。

注: このサンプルはOracle WebLogic Server 12.2.1.4または14.1.1.0のドメインの作成と実行で使用できます。Dockerfileを編集し、上記の手順でビルドしたイメージを指示するように“FROM”句を変更してください。例：“FROM oracle/weblogic:12.2.1.4-dev”

/OracleWebLogic/samples/12213-domainおよび/OracleWebLogic/samples/12213-domain/container-scriptsサブディレクトリ内にサポート・スクリプトがあります。これらは上記のDockerfileとともに、Oracle WebLogic Server ドメイン・イメージをビルドし、ドメインを構成し、管理サーバーおよびクラスタ内の管理対象サーバーを実行するために必要です。

スクリプト名	機能
build.sh	“12213- weblogic-domain-in-volume”という名称のOracle WLS ドメイン・イメージをビルドします。
run-admin-server.sh	properties ファイルから資格証明を取得し、boot.properties ファイルを作成し、create-wls-domain.pyを呼び出してOracle WLSTを実行してドメイン構成を作成し、管理サーバーを起動します。

run-managed-server.sh	1台の管理サーバーを含むベース・ドメインを構成します。
properties/domain.properties	以下の値が含まれており、これらを使用してドメインをカスタマイズできます。 DOMAIN_NAME=myDomain ADMIN_PORT=7001 ADMIN_NAME=myadmin ADMIN_HOST=AdminContainer MANAGED_SERVER_PORT=8001 MANAGED_SERVER_NAME_BASE=MS CONFIGURED_MANAGED_SERVER_COUNT=2 PRODUCTION_MODE_ENABLED=true CLUSTER_NAME=cluster1 CLUSTER_TYPE=DYNAMIC DOMAIN_HOST_VOLUME=/Users/host/temp
properties/domain_security.properties	Oracle WebLogic管理サーバーの資格証明です。
container-scripts/setEnv.sh	上記のパラメータを設定してサーバーを実行します。
container-scripts/createWLSDomain.sh	domain_security.propertiesファイルから資格証明を取得し、boot.propertiesファイルを作成し、create-wls-domain.pyを呼び出してOracle WLSTを実行し、ドメイン構成を作成します。
container-scripts/create-wls-domain.py	1台の管理サーバー、クラスタ（構成済みまたは動的）、クラスタ内の管理対象サーバーを含むドメインを構成します。
container-scripts/startAdminServer.sh	管理サーバーを起動します。
container-scripts/startManagedServer.sh	クラスタ内の特定の管理対象サーバーを起動します。

Oracle WebLogic Server ドメインのサンプルDockerイメージのビルド

ドメインが構成されたOracle WebLogic Serverイメージを作成するには、以下の手順を実行します。

Docker GitHubリポジトリのクローンをローカル・マシンに作成します（まだ作成していない場合）。このホワイト・ペーパーで示されるサンプル・ファイルには、これらのDockerfileとスクリプトが必要になります。

```
$ git clone https://github.com/oracle/docker-images.git
```

上記の説明に従ってoracle/weblogic:12.2.1.3-devイメージをビルドするか、あらかじめビルドされたイメージを[Oracle Container Registry](#)から取得しておいてください。初めて使用するユーザーは、こちらの[説明](#)に従ってください。

Oracle Container Registryからイメージを取得するコマンドは以下のとおりです。

```
$ docker login container-registry.oracle.com  
$ docker pull container-registry.oracle.com/middleware/weblogic:12.2.1.3-dev  
$ docker tag container-registry.oracle.com/middleware/weblogic:12.2.1.3-dev  
oracle/weblogic:12.2.1.3-dev
```

OracleWebLogic/samples/12213-domainで、以下のコマンドを実行します。

```
$ build.sh  
または  
$ docker build -f Dockerfile -t 12213-weblogic-domain-in-volume.
```

このイメージが作成されていることを確認するには、次のコマンドを実行します。

```
$ sudo docker images
```

Oracle WebLogic Server管理サーバー・コンテナの実行

Oracle WebLogic Server ドメイン・イメージを実行してコンテナを起動すると、デフォルトでは、そのコンテナで管理サーバーが稼働を開始します。デフォルトの管理サーバー名は"AdminServer"で、デフォルトのポート番号は7001、デフォルトの管理サーバー・コンテナ名は"wlsadmin"です。

注：同じ1つのホスト内で複数のドメインを実行する場合は、サーバーが稼働中のDockerポート番号とコンテナ名を変更する必要があります。

ホスト・ボリュームにWebLogic Server ドメインを作成する場合は、**docker run**コマンドを実行するだけで作成できます。“12213-weblogic-domain-in-volume”がOracle WebLogic Server ドメイン・イメージのタグです。サンプルのDockerfileでは、**container-scripts/createWLSDomain.sh**がデフォルトのCMD（コマンド）として定義されています。

上記の表で説明したとおり、**properties/domain_security.properties**ファイルにOracle WebLogic Serverの資格証明を入力し、**properties/domain.properties**にドメインのプロパティを定義します。

このサンプルでは、ホスト・ボリューム上にドメイン・ホームが作成されます。docker runコマンド内で、ホスト・ボリュームをプロパティ **-v \${domainhostvol}:/u01/oracle/user_projects/domains** にマッピングしてください。ディレクトリ **\${domainhostvol}** はファイル**properties/domain.properties**のDOMAIN_HOST_VOLUMEで定義されます。

管理サーバーを起動するには、以下のコマンドを実行します。

```
$ run-admin-server.sh
```

または

```
$ container-scripts/setEnv.sh
```

```
$ sudo docker run -d -p 9001:7001 -p 9002:9002 --name ${adminhost}           |
--hostname ${adminhost} -v ${scriptDir}/properties:/u01/oracle/properties           |
-v ${domainhostvol}:/u01/oracle/user_projects/domains                           |
${ENV_ARG} 12213-weblogic-domain-in-volume
```

管理サーバー・コンテナのIPアドレスを取得するには、以下のコマンドを実行します。

```
$ sudo docker inspect --format '{{.NetworkSettings.IPAddress}}' ${adminhost}
```

戻り値の例：**xxx.xx.x.xx**

これで、Oracle WebLogic Server管理コンソールに次のURL（<http://xxx.xx.x.xx:7001/console>）でアクセスできるようになりました。

注：同じホスト上で複数のOracle WebLogic Serverドメインを稼働させている場合（管理サーバーが複数ある場合）は、**-name**（管理サーバー・コンテナの名前）と**-p**（管理サーバーのポートがマッピングされているDockerポート）を変更してください。

Oracle WebLogic Server管理対象サーバー・コンテナの実行

container-scripts/createWLSDomain.shの実行によってドメインが作成されると、定義された数（**CONFIGURED_MANAGED_SERVER_COUNT**）の管理対象サーバーが設定されたクラスタがクラスタ内に構成されます。ドメイン内の管理対象サーバーのそれぞれに一意の名前が付けられ、その名前は基本名（**MANAGED_SERVER_NAME_BASE**）に番号を付けて生成されます。すべての管理対象サーバーはポート8001でリスニングします。

管理対象サーバー・コンテナは、管理サーバー・コンテナの名前\${adminhost}を使用してリンクする（**--link**コマンド）ことにより、管理サーバー・コンテナと通信します。管理サーバー・コンテナの名前はデフォルトで“wlsadmin”になります。複数のドメインが同じホスト上で稼働している場合は、管理サーバー・コンテナ名を一意にする必要があります。**-name \${adminhost}** パラメータを使用して、管理サーバー・コンテナ名を、各管理対象サーバー・コンテナの**-link**コマンドで指定した名前と一致するよう変更してください。

管理対象サーバーを起動するには、以下のスクリプトを実行します。

```
$run-managed-server.sh <managed-server-name> <maped-port>
```

または

```
$ container-scripts/setEnv.sh
```

```
$ sudo docker run -d -p ${maped-port}:${CUSTOM_MANAGED_SERVER_PORT} -v ${scriptDir}/properties:/u01/oracle/properties ${ENV_ARG} --volumes-from {adminhost} --link ${adminhost}:${adminhost} --name ${managed-server-name}
```

|

| 12

```
213-weblogic-domain-in-volume /u01/oracle/container-scripts/startManagedServer.sh
```

注：管理対象サーバー・コンテナがすでに稼働しているホストOSに追加の管理対象サーバー・コンテナを作成する場合は、新しい一意のDockerリスニング・ポート（-p）を割り当てる必要があります。このようにすると、同じホストOSで稼働する複数の管理対象サーバーが同じリスニング・ポートでリスニングすることがなくなります。たとえば、Managed Server1ではプロパティ-p 9810:8100を使用してポートをマッピングし、Managed Server2ではプロパティ-p 9820:8100を使用してポートをマッピングします。

その他のサンプル

これまでのサンプルは、Oracle WebLogic Serverドメインを共有ホスト・ボリューム上に構成するものでした。その他に、イメージ内にOracle WebLogicドメイン・ホームを作成する方法を示すサンプルが2つあります。OracleWebLogic/samples/12213-domain-home-in-imageサンプルはOracle WLSTを使用してイメージ内にドメイン・ホームを作成し、OracleWebLogic/samples/12213-domain-home-in-image-wdtサンプルはOracle WDTを使用してイメージ内にドメイン・ホームを作成します。Oracle WebLogic Deploy Toolingについて詳しくは、[Oracle WebLogic Deploy ToolingのGitHub](#)プロジェクトを参照してください。

Oracle WebLogic ServerをDockerで実行する場合の考慮事項

- Dockerコンテナが再起動されてIPアドレスが変更されると、そのDockerコンテナ内で稼働するOracle WebLogic Serverのアドレスが新しいものになります。コンテナの再起動前にそのサーバーと通信していたアプリケーションや他のサーバーは通信できなくなります。このコンテナ再起動後のIPアドレス変更に対しては、DockerでDNSサーバーを構成し、DNS名を使用するようにOracle WLSドメインを構成することが解決策となります。
- Oracle WebLogic Server構成、サーバー・ログ、ファイル・ストアなどは、すべてコンテナのファイル・システムに格納されます。Dockerコンテナが破損すると、ファイル・システム全体が失われます。代わりとなる方法は次の2つです。
 - ホストのファイル・システムを使用してコンテナのローカル・ファイル・システムを保存する。
 - “データ専用”のコンテナを用意して、ドメインのファイル・システムを格納する。なるべくファイル・システムに依存しないようにするために、以下のようにすることをお薦めします。
 - TLog、JMSなどのストアはデータベースに保持する。

- XAトランザクションを使用する場合は、“TLog書き込みなしのXAトランザクション”を使用して、TLogへの書き込みを最小限に抑える。
- クラスタ化されたOracle WebLogic Serverは、互いに通信し、管理サーバーとも通信する必要があります。異なるホスト・マシンで稼働するDockerコンテナには、他のコンテナと直接通信するために必要になる可視性やアクセス権がありません。そのため、現時点では、複数のホストのオペレーティング・システムにまたがるOracle WebLogic Server構成の使用はサポートされていません。これに代わる構成としては、Oracle WebLogic Server ドメイン全体を単一のホストで運用するという方法があります。
- WebLogic Serverイメージに最新のPatch Set Updates (PSU) のパッチを適用することをお薦めします。これらのパッチには、本番環境のデプロイメントを保護するために必要になる重要なセキュリティ修正プログラムが含まれています。Oracle WebLogic Serverイメージへのパッチ適用には、以下の2つの方法があります。

 - Oracle WebLogic Image Toolを使用すれば、Oracle WebLogic Server用のDockerイメージの作成とパッチ適用を簡素化し、自動化することができます。このツールにより、JDKとOracle WebLogic Serverのインストールを含む新しいLinuxベースのイメージを作成し、さらにオプションで、Oracle WebLogic ドメインとアプリケーションの構成、Oracle WebLogic Serverパッチの適用、または既存イメージの更新を実行できます。ドキュメントとサンプルについては、[Image ToolのGitHub](#)リポジトリを参照してください。
 - また、手動でOracle WebLogicイメージにパッチを適用することもできます。そのためには、Oracle WebLogic Server汎用インストール、開発者向けインストール、またはスリム・インストールのイメージにパッチを適用するためのDockerfileを作成します。このサンプルについては、DockerのGitHubリポジトリ (OracleWebLogic/samples/12213-patch) を参照してください。

- DockerコンテナおよびLinuxコンテナのセキュリティに関して、次の懸念事項があります。

 - 1つには、別々のコンテナで実行中のコードを分離させられるかどうかという懸念があります。そのような環境でOracle WebLogic Serverを実行できなくなるような問題は、現時点では確認されていません。
 - 他に、Dockerイメージの提供元に関する懸念があります。Dockerイメージは信頼できる提供元のみから取得する必要があります。また、更新頻度とDocker Hubの管理特性を意識する必要があります。
 - DockerおよびLinuxテクノロジーの最新情報を絶えず入手し、それぞれで発生しているセキュリティ上の問題を常に認識しておく必要があります。
 - Dockerコンテナのデフォルトのネットワーク・モードである“ブリッジ・ネットワーク”では、マルチキャストがサポートされません。Dockerコンテナの“ホスト・ネットワーク”ではマルチキャストがサポートされますが、ホスト・ネットワークのスタッフが使用されるため、独立性が低下します。DockerコンテナでOracle WebLogic Serverクラスタを実行する場合は、クラスタ・プロトコルとしてユニキャストを使用することをお薦めします。

まとめ

Dockerのアーティファクトは移植が可能で複数のLinux環境への分散が容易であるため、Dockerは運用の簡素化とコストの削減を実現するテクノロジーとして有望です。

オラクルはお客様の関心の高まりに応え、DockerコンテナでのOracle WebLogic Serverの動作を保証し、Dockerコンテナで動作するOracle WebLogic Server構成の作成を支援するイメージ、Dockerfile、スクリプトを提供してきました。これらがお客様のお役に立つことを願っています。オラクルは今後も、Docker環境のサポート範囲の向上を目指していきます。



Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

海外からのお問い合わせ窓口

電話 : +1.650.506.7000
ファクシミリ : +1.650.506.7200

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。0420

DockerコンテナでのOracle WebLogic Serverの運用
2020年4月
WebLogic Server PM、Monica Riccelli

 | Oracle is committed to developing practices and products that help protect the environment